AD-A090 772 WASHINGTON UNIV SEATTLE DEPT OF COMPUTER SCIENCE F/G 12/1
OPTIMAL RESOURCE PLACEMENT IN A DISTRIBUTED SYSTEM. (EXTENDED A--ETC(U)
AUG 80 M J FISCHER, N D GRIFFETH, N A LYNCH N00014-80-C-0221
UNCLASSIFIED TR-80-08-03 NL

1 OF 1
AD A 090772

END
DATE FILMED
11-80
DTIC

LEVEL II

12

OPTIMAL RESOURCE PLACEMENT IN A DISTRIBUTED SYSTEM*
(extended abstract)

Michael J. Fischer
University of Washington
Seattle, WA 98195

Nancy D. Griffeth and Nancy A. Lynch
Georgia Institute of Technology
Atlanta, GA 30332

Technical Report 80-08-03 August 1980

*This material is based upon research supported by the Office of Naval Research under Contracts N00014-80-C-0221 and N0014-79-C-0873, and by the U.S. Army Research Office Contract Number DAAG29-79-C-0155.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1 REPORT NUMBER: TR-80-08-03	2 GOVT ACCESSION NO: AD-A090 172
3 RECIPIENT'S CATALOG NUMBER:	
4 TITLE (and Subtitle): OPTIMAL RESOURCE PLACEMENT IN A DISTRIBUTED SYSTEM, (Extended Abstract),	5 TYPE OF REPORT & PERIOD COVERED: Technical Report
	6 PERFORMING ORG. REPORT NUMBER:
7 AUTHOR(s): Michael J. Fischer, ~~University of Washington~~ Nancy D. Griffeth ~~and~~ Nancy A. Lynch, Georgia Institute of Technology	8 CONTRACT OR GRANT NUMBER(s): ONR Contracts: N00014-80-C-0221 & N00014-79-C-0873; U.S. A.R.O. Con.DAAG29-79-C-0155.
9 PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Computer Science, FR-35 University of Washington Seattle, WA 98195	10 PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 049-456/30 Oct 79 (437)
11 CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research 800 North Quincy Street Arlington, VA 22217 — U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709	12 REPORT DATE: August 1980
	13 NUMBER OF PAGES: 11
14 MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research 800 North Quincy Street Arlington, VA 2-217 Attn: Dr. R. B. Grafton	15 SECURITY CLASS (of this report): Unclassified
	15a DECLASSIFICATION DOWNGRADING SCHEDULE:

16 DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)

Resource allocation, distributed system, networks.

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

We consider the problem of locating t resources on the nodes of a complete binary tree of n leaves so as to minimize the expected total distance from each of t simultaneous random requests at the leaves to the resource with which each is optimally matched. The optimal placement of resources yields an expected total distance of at most $ct + 2\log(n/t)$. Such a placement can be found in time $O(\log n)$.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102 LF 014 6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1. Introduction

We consider a problem of resource allocation in a network of computers. Imagine an n node undirected graph in which the nodes represent processors and the edges communication paths. We have t resources which can be placed on various nodes of the graph. Requests for the resources originate with equal probability at any of a designated set of request nodes. Each request is serviced by matching it with some resource. The cost of that service is the shortest distance between the request and the resource with which it is matched. The cost of servicing a simultaneous set of requests is the minimum total cost that can be obtained over all possible matchings of those requests to resources.

For example, the resources might be available processors and the requests be users wishing service. The minimum cost matching gives the best way of assigning processors to users so as to minimize average distance in the network between the user and his processor.

Averaging over all possible patterns of a fixed number of requests at the designated request points, we get an expected cost of servicing the requests. We are interested in how to place the resources so as to minimize the expected cost of servicing a random set of requests of a fixed size, and we wish to get good bounds on that expected cost.

Our results to date are for the special but interesting case with the following restrictions:

(i) The graph is a complete binary tree with n leaves;

(ii) Resources can be placed anywhere in the tree, but requests originate only at leaves;

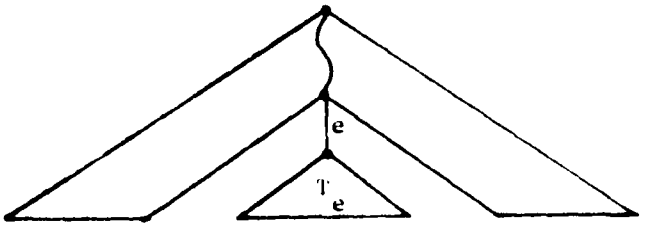
(iii) Requests are independent random events uniformly distributed over the leaves;

(iv) The number of requests equals the number of resources.

2. Properties of Optimal Placements

An assignment of t resources to nodes is optimal if it minimizes the expected cost of servicing a random set of t requests at the leaves. Intuitively, if t is small relative to n, the number of leaves in the tree, it should be better to place the resources up toward the root, whereas if t is large relative to n, then one should do better by dividing up the resources equally among the leaves. In fact, this is not strictly true unless t is a power of 2. The structure of optimal placement for non-power-of-two numbers of resources is rich and complicated; nevertheless, an optimal placement can be found quickly by a straightforward algorithm. We develop those results in the remainder of this section.

We begin by looking at the optimal matching of a fixed set of t requests to the t resources. A matching is a set of directed paths from distinct requests to distinct resources. The directed flow along an edge e in the tree is the number of paths in the matching which contain that edge in the designated direction (towards or away from the root). Because we are looking only at trees, a necessary and sufficient condition for a matching to be optimal is that no edge have flow in both directions. It follows that the flow along any edge e in an optimal matching is determined solely by the number of requests in the subtree T_e which e connects to the rest of the tree, as shown below:



If the number of requests in T_e exceeds the number of resources in T_e, then the difference flows through e out of T_e. If the number of resources exceeds the number of requests, then the difference flows through e into T_e. The flow is zero if the number of requests equals the number of resources in T_e. Letting s be the number of resources and i the number of requests in T_e, the absolute flow is defined to be $|s-i|$. Summing the absolute flow over all edges in the tree gives the cost of the optimal matching.

We now consider all the different possible patterns of requests and look at the expected absolute flow along an edge e. If T_e has s resources, this is given by

$$\sum_{i=0}^{t} \text{Prob[exactly } i \text{ requests occur in } T_e] \cdot |s-i|.$$

Since the requests are independent and uniformly distributed, the probability of exactly i requests in a subtree is binomially distributed and is given by $\binom{t}{i} p_e^{\,i} \cdot q_e^{\,t-i}$, where t is the total number of requests, p_e = (# leaves in T_e)/n is the probability that a given request will fall in T_e, and $q_e = 1-p_e$. Thus, the expected absolute flow in e is given by

$$f_e(s) = \sum_{i=0}^{t} \binom{t}{i} p_e^{\,i} \cdot q_e^{\,t-i} \cdot |s-i|.$$

Let E_e be the set of edges in the subtree T_e, including e itself. For each edge d in E_e, let s_d be the number of resources in the subtree T_e. The total expected absolute flow in T_e is defined to be

$$F_{E_e}(\underset{\sim}{s}) = \sum_{d \in E_e} f_d(s_d),$$

where $\underset{\sim}{s}$ is the vector $(s_d)_{d \in E_e}$.

Note that F_{E_e} includes the flow along the edge leading out of the root of the subtree as well as the flows on all the edges internal to the subtree.

The definitions for f_e and F_{E_e} make perfectly good mathematical sense for arbitrary real values of s. This corresponds to the situation in which resources are continuously divisable, but the requests nevertheless are integral. We consider the continuous case for technical convenience.

Let the optimal flow in the continuous case be given by

$$F^{opt}_{E_e}(s) = \min \{F_{E_e}(\underset{\sim}{g}) \mid s = \sum \underset{\sim}{g}, \text{ and each } s_d \text{ in } \underset{\sim}{g} \text{ is a non-negative real number}\}$$

For the integer case, we have

$$F^{opt}_{E_e}(s) = \min \{F_{E_e}(\underset{\sim}{g}) \mid s = \sum \underset{\sim}{g}, \text{ and each } s_d \text{ in } \underset{\sim}{g} \text{ is in } \mathbf{N}\}.$$

$\hat{F}^{opt}_{E_e}$ is defined only for $s \in \mathbf{N}$.

Theorem 1. Each of the functions f_e, F_{E_e}, and $F^{opt}_{E_e}$ is convex, piecewise linear, with vertices occuring at integer values of s.

Theorem 2. $F^{opt}_{E_e}(s) = \hat{F}^{opt}_{E_e}(s)$ for all $s \in \mathbf{N}$.

This tells us that the freedom to subdivide resources cannot result in a lower cost placement than can be achieved by placing whole resources.

The fair share placement of t resources, which plays a special role in our development, puts t/n resource units on each leaf and no resources on any internal node. This placement has the property that the fraction of resources on any subtree is the same as the fraction of leaves which that

subtree contains. One might conjecture that the fair share placement is optimal, and indeed it is when t/n is an integer, but there are cases where it is demonstrably non-optimal. Nevertheless, it is always close to optimal, a fact that is crucial to the correctness of our algorithm.

Theorem 3. There exists an optimal placement (of whole resources) such that (i) for every pair of brother subtrees T_e and $T_{e'}$, the numbers of resources on T_e and $T_{e'}$ differ by at most 1;

(ii) the number of resources on each subtree T_e differs from the fair share number for that subtree by less than one.

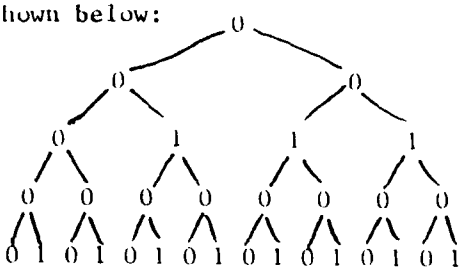
The proof of this theorem rests heavily on Theorems 1 and 2 and uses the little-known fact that the mean and the median of the binomial distribution differ by an amount less than one [1,2].

Corollary. If the number of resources is a multiple of the number of leaves, then the fair share placement is optimal.

From Theorem 3, we get a time $O(\log n)$ dynamic-programming algorithm for finding an optimal placement of t resources on a complete binary tree of n leaves. The algorithm proceeds by finding, for each $L = 0,1,\ldots,$ $\log n$ the cost of the optimal placement of both $\lfloor h \rfloor$ and $\lceil h \rceil$ resources on a subtree of 2^L leaves, where $h = t \cdot 2^L/n$ is the fair share number for that subtree. This takes only a constant amount of time given the same information for $L-1$; hence the total time is $O(\log n)$. A "description" of the actual placement can also be found in time $O(\log n)$, but of course to produce the fully-written-out placement takes time $O(n)$. This algorithm establishes:

Theorem 4. The cost of an optimal placement of t resources on a complete binary tree of n leaves, and a "description" of that placement, can be computed in time $O(\log n)$ on a RAM.

We conclude this section with an example showing that the fair share placement is not always optimal and illustrating some of the structure of optimal placements. An optimal placement of 11 resources on a 16-leaf complete binary tree is shown below:



The expected cost of this placement is 24.902... . The fair share placement, which places 0.6875 resources on each leaf, has expected cost 25.404... .

3. Bounds on the Costs of Optimal Layouts

In this section, we show that the optimal placement of t resources results in an expected cost of at most

$$c \cdot t + 2 \log \frac{n}{t}$$

where c is a "small" constant. Hence, the average cost per request is constant when the number of resources is at least proportional to the number of leaves in the tree.

We begin by defining a simple placement to be the one which puts $t/2^L$ resources on each node at level L (from the root), where $L = \min(\log n, \lceil \log t \rceil)$. Note that when $t > n/2$, the simple and fair share placements are the same. Also, no node in a simple placement has a non-zero number of resources $< \frac{1}{2}$.

Theorem 5. Cost of the simple placement $\leq$ cost of the fair share placement $\leq c \cdot t + 2 \log \frac{n}{t}$ for some constant c.

Theorem 6. $F^{opt}(t) \leq$ cost of simple placement $\leq 2L - 1 + F^{opt}(t)$ where $L = \min\{\log n, \lceil \log t \rceil\}$.

Theorem 7. If t is a power of 2, the simple placement is optimal.

4. Discussion

A related problem, which formed the original motivation for this study, was to find and analyze algorithms for a distributed "ticket" system such as might be used to sell tickets to a sporting event. The tickets are assumed to reside at various nodes in the network, and the distance from a request to the ticket which is eventually used to grant the request is some measure of the waiting time to service that request. The difference between that problem and the one in this paper is that the requests come in sequentially and it is not possible to achieve an optimal matching of requests to tickets without knowing the locations of future requests. We are currently trying to extend our results to this case of serial requests for resources.

It seems clear that any static placement of tickets on nodes will lead to deteriorating performance as more and more tickets are sold. Thus, even though the average waiting time might be acceptably small, the time might nevertheless grow unacceptably large when only few tickets are left. To help alleviate this problem, one might consider algorithms which dynamically move tickets around to try to improve the future expected waiting time, analogous to the various balancing schemes for binary search trees. We have an algorithm which does such rebalancing, but it is still unclear to what extent, if any, the rebalancing actually helps.

Finally, it would be nice to remove some of the technical restrictions we placed on the problem and generalize the results to cases of arbitrary graphs, arbitrary subsets of nodes for location of requests and placement of resources, numbers of requests different from number of resources, and non-uniform distributions on the probability of a request.

References

1. Norman L. Johnson and Samuel Kotz, Distributions in Statistics, Vol. 1, "Discrete Distribution," John Wiley & Sons, Inc., New York, 1969.

2. Von W. Uhlmann, "Vergleich der hypogeometrischen mit der Binomial-Verteilung," Metrika 10 (1966), 145-148.

DISTRIBUTION LIST

Office of Naval Research Contract N00014-80-C-0221
Michael J. Fischer, Principal Investigator

Defense Documentation Center
Cameron Station
Alexandria, VA 22314
(12 copies)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. R. B. Grafton, Scientific Officer (1 copy)
Information Systems Program (437) (2 copies)
Code 200 (1 copy)
Code 455 (1 copy)
Code 458 (1 copy)

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106
(1 copy)

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375
(6 copies)

Office of Naval Research
Resident Representative
University of Washington, JD-27
422 University District Building
1107 NE 45th Street
(1 copy)

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380
(1 copy)

Naval Ocean Systems Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152
(1 copy)

Mr. E. H. Gleissner
Naval Ship Research and Development Center
Computation and Mathematics Dept.
Bethesda, MD 20084
(1 copy)

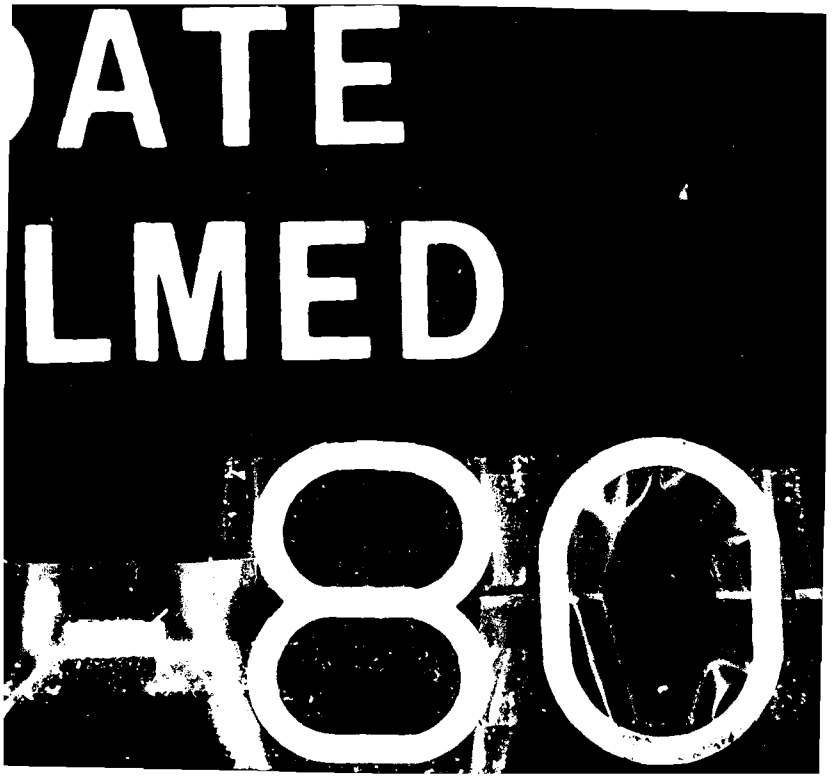
Captain Grace M. Hooper (008)
Naval Data Automation Command
Washington Navy Yard
Building 166
Washington, D.C. 20374
(1 copy)

Defense Advanced Research Projects Agency
Attn: Program Management/MIS
1400 Wilson Boulevard
Arlington, VA 22209
(3 copies)

Professor Nancy A. Lynch
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

Professor Philip Enslow
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, IL 60605


ATE
LMED
80